CLAIMS

What is Claimed is:

1. A method comprising:

hooking an exception handler dispatcher;

stalling execution of said exception handler dispatcher upon invocation of said exception handler dispatcher; and

determining whether an exception handling is valid, wherein upon a determination that said exception handling is valid, said method further comprising allowing said execution of said exception handler dispatcher to proceed.

2. The method of Claim 1 wherein said determining whether an exception handling is valid comprises:

determining whether exception handler frame addresses are in order.

3. The method of Claim 2 wherein said determining whether exception handler frame addresses are in order comprises determining whether said exception handler frame addresses are successively increasing from a first exception handler frame located highest on a stack.

4. The method of Claim 1 wherein said determining whether an exception handling is valid comprises:

determining whether an exception handler is in a data area of memory.

5. The method of Claim 4 wherein said determining whether an exception handler is in a data area of memory comprises determining whether a handler address in an exception handler frame points to a page in said data area.

6. The method of Claim 1 wherein said determining whether an exception handling is valid comprises:

determining whether a previous exception handler frame address is invalid.

7. The method of Claim 6 wherein said determining whether a previous exception handler frame address is invalid comprises determining whether said previous exception handler frame address in an exception handler frame points to a page that is invalid.

8. The method of Claim 1 wherein exception handler frames form a linked list, said determining whether an exception handling is valid comprises:

determining whether exception handler frame addresses of said exception handler frames are in order.

9. The method of Claim 8 wherein said determining whether exception handler frame addresses of said exception handler frames are in order comprises determining whether said exception handler frame addresses are successively increasing from a first exception handler frame located highest on a stack, said linked list comprising said first exception handler frame.

10. The method of Claim 1 wherein exception handler frames form a linked list, said determining whether an exception handling is valid comprises:

determining whether any exception handlers associated with said exception handler frames are in a data area of memory.

11. The method of Claim 10 wherein said determining whether any exception handlers associated with said exception handler frames are in a data area of memory comprises determining whether any handler addresses in said exception handler frames point to a page in said data area.

12. The method of Claim 1 wherein exception handler frames form a linked list, said determining whether an exception handling is valid comprises:

determining whether any previous exception handler frame addresses in said exception handler frames are invalid.

13. The method of Claim 12 wherein said determining whether any previous exception handler frame addresses in said exception handler frames are invalid comprises determining whether said any previous exception handler frame addresses in said exception handler frames point to a page that is invalid.

14. The method of Claim 1 wherein upon a determination that said exception handling is not valid during said determining, said method further comprising taking protective action.

15. The method of Claim 14 wherein prior to said taking protective action, said method further comprising determining that said exception handling is not a known false positive exception handling.

16. The method of Claim 14 further comprising providing a notification that said protective action has been taken.

17. The method of Claim 1 wherein said hooking comprises hooking a function called KiUserExceptionDispatcher().

18. The method of Claim 1 wherein said hooking comprises modifying said exception handler dispatcher to redirect flow to an exception handling validation module.

19. The method of Claim 18 wherein said modifying comprises inserting a jump instruction into said exception handler dispatcher.

20. The method of Claim 1 further comprising invoking said exception handler dispatcher, said invoking comprising raising an exception.

21. A method comprising:

determining that exception handling is valid prior to allowing execution of an exception handler dispatcher.

22. A computer program product comprising:

an exception handling validation application for hooking an exception handler dispatcher;

said exception handling validation application further for stalling execution of said exception handler dispatcher upon invocation of said exception handler dispatcher; and

said exception handling validation application further for determining whether an exception handling is valid, wherein upon a determination that said exception handling is valid, said exception handling validation application further for allowing said execution of said exception handler dispatcher to proceed.

23. The computer program product of Claim 22 wherein said determining whether an exception handling is valid comprises:

determining whether exception handler frame addresses are in order.

24. The computer program product of Claim 22 wherein said determining whether an exception handling is valid comprises:

determining whether an exception handler is in a data area of memory.

25. The computer program product of Claim 22 wherein said determining whether an exception handling is valid comprises:

determining whether a previous exception handler frame address is invalid.